(19) Canadian Intellectual Property Office
An Agency of Industry Canada

Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

(11) CA 2 231 992 (13) A1
(43) 13.09.1999

(12)

(21) 2 231 992

(51) Int. Cl.^{6}: **H04L 029/06**, H04L 012/24

(22) 13.03.1998

(71)
Newbridge Networks Corporation,
600 March Road
P.O. Box 13600, KANATA, O1 (CA).

(72)
MULLER, Robert (CA).
BURNS, John (CA).
TOOKER, Mark (CA).
BOSLOY, Jonathan (CA).

(74)
Marks & Clerk

(54) PROTOCOLE DE GESTION DE RESEAUX POUR RECHERCHES EFFICACES
(54) NETWORK MANAGEMENT PROTOCOL FOR EFFICIENT RETRIEVAL OPERATIONS

(57)

A network management protocol for the efficient retrieval of items of information or objects from managed elements or other network management systems in a digital communications network is described. The new protocol introduces one or more qualifier bits appended to an information retrieval command wherein the qualifier bit specifies the subclasses of the objects requested.

OPIC
Office de la propriété
intellectuelle du Canada

CIPO
Canadian Intellectual
Property Office

(12) (19) (CA) **Demande-Application**

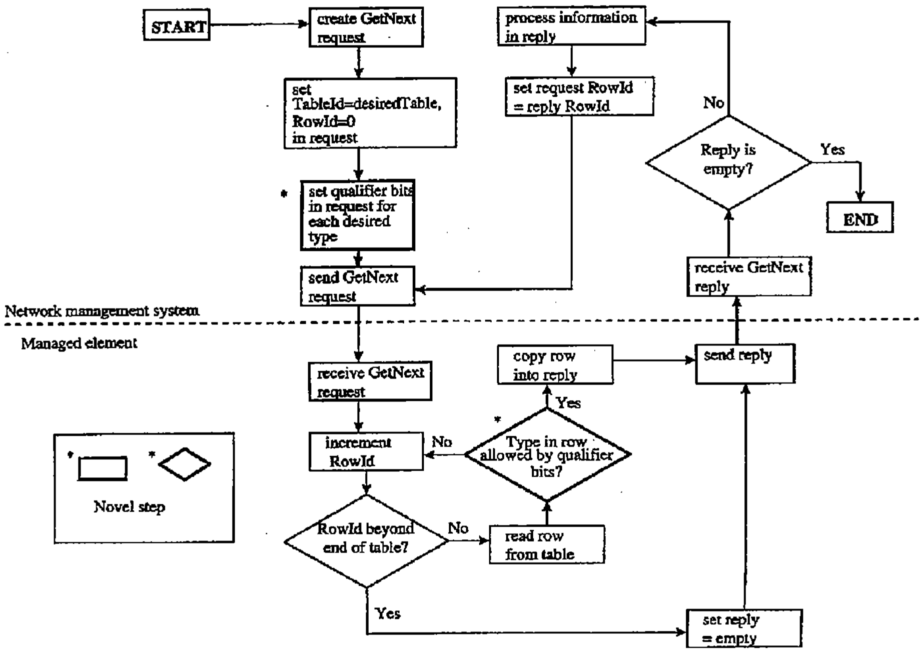
(21) (A1) **2,231,992**
(22) 1998/03/13
(43) 1999/09/13

(72) TOOKER, Mark, CA
(72) BOSLOY, Jonathan, CA
(72) BURNS, John, CA
(72) MULLER, Robert, CA
(71) Newbridge Networks Corporation, CA

(51) Int.Cl.^{6} H04L 29/06, H04L 12/24

(54) **PROTOCOLE DE GESTION DE RESEAUX POUR RECHERCHES EFFICACES**

(54) **NETWORK MANAGEMENT PROTOCOL FOR EFFICIENT RETRIEVAL OPERATIONS**


START
create GetNext request
set TableId=desiredTable, RowId=0 in request
set qualifier bits in request for each desired type
send GetNext request
process information in reply
set request RowId = reply RowId
No
Reply is empty?
Yes
END
receive GetNext reply
Network management system
Managed element
receive GetNext request
increment RowId
No
RowId beyond end of table?
No
read row from table
Yes
set reply = empty
Type in row allowed by qualifier bits?
Yes
copy row into reply
send reply
Novel step

(57) A network management protocol for the efficient retrieval of items of information or objects from managed elements or other network management systems in a digital communications network is described. The new protocol introduces one or more qualifier bits appended to an information retrieval command wherein the qualifier bit specifies the subclasses of the objects requested.

Abstract

A network management protocol for the efficient retrieval of items of information or objects from managed elements or other network management systems in a digital communications network is described. The new protocol introduces one or more qualifier bits appended to an information retrieval command wherein the qualifier bit specifies the subclasses of the objects requested.

Network management protocol for efficient retrieval operations

Field of the Invention

This invention relates to communications network management systems and more particularly to a protocol for improving the retrieval of specified items of information from network management systems and managed elements.

Background

Protocols used by communications Network Management Systems (NMSs) for retrieving information from managed elements (MEs) or other NMSs commonly define a "getnext" operation. The getnext operation is used in order to retrieve one or more items of information (called objects), when the NMS does not know beforehand how many objects exist, or even if any objects exist. The NMS typically sends repeated getnext commands to retrieve all objects contained in a series (such as all rows in a logical table), starting at the beginning of the series and continuing until either no more reply is received to the getnext command, or else a reply is received containing an object which lies outside of the desired series.

Similarly, a "getbulk" command is typically defined, which allows an NMS to simultaneously retrieve all items in a series, by specifying (a) the object at the start of the desired series, and (b) the maximum number of successive objects to be retrieved.

The shortcoming of these approaches occurs when the same objects may be viewed as belonging to different series simultaneously. For example, a ME that is a packet switching device may have a table of cross connection objects. Within the table, there are types of cross connections, such as Permanent Virtual Circuits (PVCs), Switched Virtual Circuits (SVCs), and Soft Permanent Virtual Circuits (SPVCs). For the purposes of accounting for total utilization of the ME, the NMS may wish to retrieve information on all cross connections on the ME, regardless of type. For the purposes of allowing a user of the NMS to view and manage only particular types of cross connections (for example, only PVCs and SPVCs), the NMS may wish to retrieve information on these types but not on any other.

Prior Art

There are two prior art solutions to the problem of retrieving only selected objects from a series.

In the first prior art solution, the NMS retrieves all items in the series, and discards all those of a type which is not desired, before processing the information further. This has the major shortcoming that it is inefficient, from the point of view of:

a) processing resources on the ME (to copy information on each object into the “get” reply);

b) processing resources on the NMS (to retrieve and examine each object from the reply); and

c) the amount of bandwidth utilized for communications between the ME and the NMS.

In the second prior art solution, different logical series of data are defined on the ME for the same information. For example, cross connections on a packet switch can be retrieved from an “all cross connections” table, from a “PVC” table, from a “SVC” table, etc. The same objects would be accessible from different tables, depending on their type and the definition of the table. The NMS can therefore select whichever table is desired, to fulfill a particular management request. The shortcoming of this approach is that it is inflexible. The NMS cannot make “ad hoc” queries, that are for new or different types or combinations of types of objects within a series, unless the providers of the NMS and ME have previously agreed upon all types of series that will be supported.

Summary of the Invention

It is an object of the present invention to provide an improved protocol for network management systems to retrieve objects from managed elements within the network or from other network managers.

Therefore in accordance with the present invention there is provided in a communications network management system in which items of information respecting managed elements within the network are retained within the managed elements, a protocol for improving retrieval of information from the managed elements, the improvement comprising adding one or more qualifiers to the items of information wherein the qualifiers represent subclasses of the information.

Brief Description of the Drawings

The invention will now be described in greater detail with reference to the attached drawings wherein:

Figure 1 shows an asynchronous transfer mode (ATM) switching system containing several virtual channel circuit (VCC) cross connections between external communications interfaces;

Figures 2A and 2B illustrate a Simple Network Manager Protocol (SNMP) management information base (MIB) tree;

Figure 3 shows an example table of VCC cross connection objects relating to the various types of connections, which would be contained within a database of managed objects maintained by the ATM switch illustrated in Figure 1;

Figure 4 shows a table of cross connection objects, derived from the table in Figure 3, as the table would appear at the NMS having been received from the ME in response to the protocol of the present invention;

Figure 5 is a flowchart illustrating the procedures followed by an NMS and ME in using an efficient getnext protocol to retrieve only selected types of rows from a table; and

Figure 6 is a flowchart illustrating the procedures followed by an NMS and ME in using an efficient getbulk protocol to retrieve only selected types of rows from a table.

Detailed Description of the Invention

The basis of this invention is a protocol together with corresponding functionality effected at a network management system (NMS) and a managed element (ME) within a communications network, whereby the NMS can flexibly define which subclasses of objects it wishes to retrieve from a given series of objects at the ME. This is accomplished through the use of one or more extra qualifiers, for example, appended to the well-known getnext or getbulk commands which the NMS sends to the ME or another NMS. The qualifiers specify which subclasses of objects are to be retrieved from the series.

An example of such a qualifier would be a bitmap field, in which a 1 (one) or a 0 (zero) in an agreed-upon bit position specifies whether objects of a particular type should be retrieved from the series. In this way, the NMS can specify any number of types of object to retrieve, by setting one or more bits in the qualifier.

The getnext and getbulk requests including the qualifier, in accordance with the invention, may be advantageously used within the context of an ATM communications network by the NMS to retrieve cross connection and external interface (e.g. port) information from respective ATM switching systems. For example, Figure 1 illustrates an ATM switch containing several virtual channel circuit (VCC) cross connections between external communications interfaces. Should the NMS wish to retrieve information on only permanent virtual channel circuit (PVCC) cross connections, it might do so by setting only the first bit of the qualifier of the bitmap when sending getnext or getbulk requests to the ATM switch. Similarly if the NMS wishes to retrieve information on the permanent, switched, or semi-permanent virtual channel circuit (PVCC, SVCC or SPVCC) cross connections, it could do so by setting the first, second, and third bits, respectively, in the qualifier to a 1 (one).

A particular implementation of this invention may be based upon the well known Simple Network Management Protocol (SNMP). A bitmapped qualifier field could be incorporated into an SNMP management information base (MIB) object identifier (OID) as one of the fields in a table identifier. For example, the following OID could specify the top of a table containing cross connection information:

1.3.6.1.2.1.10

Without this invention, the NMS to retrieve individual rows of information from the table using getnext requests would send the first request containing an OID as follows:

1.3.6.1.2.1.10.0

The final field in the OID, “0”, specifies that the first row in the table should be retrieved. The NMS would have to send the request, examine the reply to see if the type of row retrieved is of the desired type, and if not, send another getnext request using the OID of the previously retrieved row as the OID specified in the getnext request, sending the second getnext request and repeating the process until a row of the desired type is retrieved.

Using this invention, the NMS could insert a qualifier field as the second last field in the OID specified in the getnext request, as follows:

1.3.6.1.2.1.10.<qualifier>.0

Using an OID in this way specifies to the managed element that the first row in the table identified by the OID “1.3.6.1.2.1.10” whose type matches one of the types specified in the <qualifier> field should be retrieved.

Figure 2A illustrates a MIB tree for the above request without the <qualifier> specified. Figure 2B illustrates the MIB tree in which an additional level is added to specify the various types or subclasses in the qualifier field.

According to this protocol the shortcomings of the first noted prior art solution are overcome because the ME does the filtering of objects from the table, before sending information to the NMS.

The shortcomings of the second noted prior art solution are overcome because the NMS can retrieve any single type or combination of types of data from a series, without having to use predetermined identifiers for each logical series.

Operation of the protocol in relation to getnext and getbulk requests, according to the invention, is illustrated in the following. Figure 3 shows an example a table stored on the ME, namely the ATM switching system in Figure 1, which table contains information about cross connections of various types. These include SVCC connections at ID 1 and ID 4, PVCC connections at ID 2 and ID 5 and SPVCC connections at ID 3 and ID 6. For this example an NMS using an efficient getnext or getbulk command as described for this invention could configure a bitmap field in the request protocol data unit (PDU) to contain three qualifier bits, one for each type of cross connection type. The protocol for accessing the cross connection table via an efficient getnext or getbulk command could define a bitmap field as follows to specify which endpoint types to retrieve: PVCC = 001, SPVCC = 010, SVCC = 100, PVCCs and SPVCCs = 011, SPVCCs and SVCCs = 110, PVCCs and SVCCs = 101, and PVCCs and SPVCCs and SVCCs = 111.

An NMS wishing to retrieve all SPVCC and SVCC connections could send a series of getnext commands as follows:

1. Request: command = getnext, tableName = connections, rowId = 0, connectionTypes = 110.

Reply: ID = 1, Type = SVCC, Endpoint 1 = A/1/1, Endpoint 2 = B/4/2

2. Request: command = getnext, tableName = connections, rowId = 1, connectionTypes = 110.

Reply: ID = 3, Type = SPVCC, Endpoint 1 = A/3/1, Endpoint 2 = B/4/0

3. Request: command = getnext, tableName = connections, rowId = 3, connectionTypes = 110.

Reply: ID = 4, Type = SVCC, Endpoint 1 = B/4/3, Endpoint 2 = C/18/0

4. Request: command = getnext, tableName = connections, rowId = 4, connectionTypes = 110.

   Reply: ID = 6, Type = SVCC, Endpoint 1 = B/4/5, Endpoint 2 = C/15/0

5. Request: command = getnext, tableName = connections, rowId = 6, connectionTypes = 110.

   Reply: NIL (contains zero cross connection endpoints)

An NMS wishing to retrieve all SPVCC and SVCC connections could alternatively send a getbulk command as follows:

```
command = getbulk, tableName = connections,
startingRowId = 1, maxRowsToRetrieve = 25,
connectionTypes = 110.
```

The table shown in Figure 4 containing all of the SPVCC and SVCC connections would be returned to the NMS in response to the aforementioned getbulk command.

Figure 5 is a flow diagram illustrating the procedures followed by a network management system and a managed element in using an efficient getnext protocol to retrieve only selected types of rows from a table. As shown in Figure 5 the first step is to create a getnext request which is followed by the establishment of the table ID which is the name of the table under which the information is stored. Also, the row identification or row ID is set as given in the previously discussed getnext commands. Next, the qualifier bits respecting the desired subclass or type is set. In the previous example, this would be the connection types 110 signifying all SPVC and SVCC connections. The getnext request is then sent by the network management system to the managed element as indicated by the dotted line between network management system and managed element in Figure 5. The getnext request is received by the managed element and the row ID is incremented as previously discussed. If the row ID is beyond the end of the table the request is forwarded back to the network management system and the set reply is indicated as being empty. If the row ID is not beyond the end of the table the new row is read from the table and the row is checked to see if it includes the type specified by the qualifier bits. If it is in the row, it is copied into the reply and forwarded to the network management system. If not, the row is incremented one additional row and rechecked. The response received by the network management system is checked to see if the reply indicates no additional information and if there is no additional information the process is ended. If, however, there is additional information the request is returned to the management element for further processing.

Figure 6 is a flowchart illustrating the procedures followed by a network management system and a managed element in using an efficient getbulk protocol to retrieve only selected types of rows from a table. Again, the process is started with the creation of the getbulk request and the table and row information is established including the number of desired rows to be covered in the request. The qualifier bits are set in the qualifier field as previously discussed. The network management system then sends the request to the managed element whereat the rows are examined and incremented in order to glean the requested information from the managed element. Upon completion the reply is returned to the network management system and the information is processed.

More qualifiers could be specified for a getnext or getbulk command retrieving cross connection information, for example another qualifier which specifies to retrieve either virtual channel circuit (VCC) cross connections or virtual path circuit (VPC) cross connections or both VCC and VPC cross connections.

Another example of objects belonging to a single class are networking interfaces. All of interfaces on an ME may be grouped together in a table according to their physical location within the ME, but for the purposes of accounting or for maintenance, the NMS may wish to retrieve information on groups of interfaces based on specific criteria such as their type, their configuration or their status.

The following is an example of an interface table for physical interfaces (ports) on a managed element.

INTERFACE TABLE

Row ID	Name	Type	Operational Status
1	A	OC3	UP
2	B	DS3	DOWN
3	C	T1	DOWN
4	D	DS3	UP
5	E	DS3	DOWN

QUALIFIER #1	BITS	
	01	GET DOWN INTERFACES
	10	GET UP INTERFACES
	11	GET UP & DOWN INTERFACES

QUALIFIER #2	BITS	
	001	GET OC3 INTERFACES
	010	GET DS3 INTERFACES
	100	GET T1 INTERFACES
	011	GET OC3 & DS3 INTERFACES
	101	GET T1 & OC3 INTERFACES
	110	GET T1 & DS3 INTERFACES
	111	GET T1 & OC3 & DS3 INTERFACES

QUAL #1 AND QUAL #2 MAY BE IN THE SAME BYTE

The following illustrates a getbulk command when it is desired to retrieve up to five operationally DOWN, DS3 interfaces in a managed element using the information contained in the above interface table.

```
GETBULK: TableId = Interface Table
         RowId = ø
         Max Rows = 5
         Status Qualifier = 01
         Type Qualifier = 010
```

The information returned from the getbulk command is given in the following table.

RETURNED TABLE

Row ID	NAME	TYPE	STATUS
2	B	DS3	DOWN
5	E	DS3	DOWN

This invention finds particular application in the management of communications networks. It can be used for NMS-to-ME, hierarchical NMS-to-NMS, and peer-level NMS-to-NMS communications.

Although a specific embodiment of the invention has been described and illustrated it will be apparent to one skilled in the art that numerous changes can be made to the basic concept. It is expected, however, that such changes will fall within the true scope of the invention as defined by the appended claims.

Claims:

1. In a communications network management system in which items of information respecting managed elements within the network are retained within said managed elements, a protocol for improving retrieval of information from said managed elements comprising: adding a qualifier representing a subclass of said information to said items of information.

2. A protocol as defined in claim 1 wherein said managed element is a device in said communications network.

3. A protocol as defined in claim 2 wherein said information to be retrieved from said device includes a series of related data.

4. A protocol as defined in claim 3 wherein said series of related data is a table of information where each element of information is of the same class, but each element may also belong to a subclass of information, where all of the elements belonging to a particular subclass are not located contiguously in said table .

5. A protocol as defined in claim 4 wherein said qualifier is a bitmap appended to a command issued by said network management system, in which one or more bits may be set by giving it a designated value of 1 (one) or 0 (zero).

6. A protocol as defined in claim 5 wherein said bit in said bitmap specifies one of said subclasses.

7. A protocol as defined in claim 5 wherein said command is a getnext command, whose reply should contain the next element of information in the specified table which belongs to a subclass whose corresponding bit is set in said qualifier.

8. A protocol as defined in claim 5 wherein said command is a getbulk command, whose reply should contain all of the elements in the specified table, up to the desired maximum number of elements, which belong to the subclasses whose corresponding bits are set in the qualifier.

9. In a communications network management system in which information objects respecting managed elements within the network are retained within said managed elements, a protocol for improving retrieval of said information objects from said managed elements comprising: adding one or more qualifiers to said information objects, wherein said qualifiers represent subclasses of said information.

10. A protocol as defined in claim 9 wherein said communications network is an asynchronous transfer mode (ATM) network.

11. A protocol as defined in claim 10 wherein said managed element is a device within said ATM network.

12. A protocol as defined in claim 11 wherein said device is a switching system having a plurality of external communications interfaces connected thereto.

13. A protocol as defined in claim 12 wherein said communication interfaces are cross connected by virtual channel circuits.

14. A protocol as defined in claim 13 wherein said virtual channel connections are switched virtual channel connections (SVCCs), permanent virtual channel connections (PVCCs), semi-permanent virtual channel connections (SPVCCs) or combinations thereof.

15. A protocol as defined in claim 14 wherein said qualifier is a bitmap appended to a command issued by said network management system, in which one or more bits may be set by giving it a designated value of 1 (one) or 0(zero).

16. A protocol as defined in claim 15 wherein said bitmap may contain a bit word made up of a combination of ones and zeros.

17. A protocol as defined in claim 16 wherein one of said qualifiers specifies one of said communication interfaces.

18. A protocol as defined in claim 16 wherein one of said qualifiers specifies one of said virtual channel connections.

19. A protocol as defined in claim 16 wherein one of said qualifiers specifies a combination of said virtual channel connections.

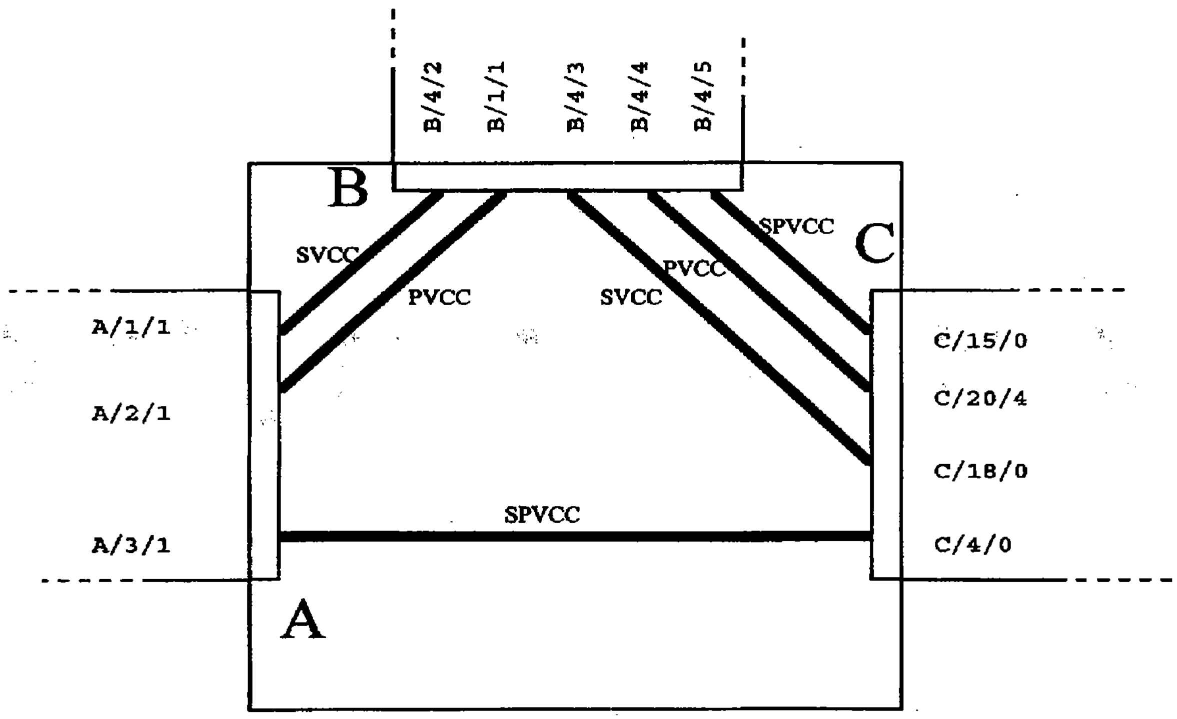


FIGURE 1

Marks & Clerk

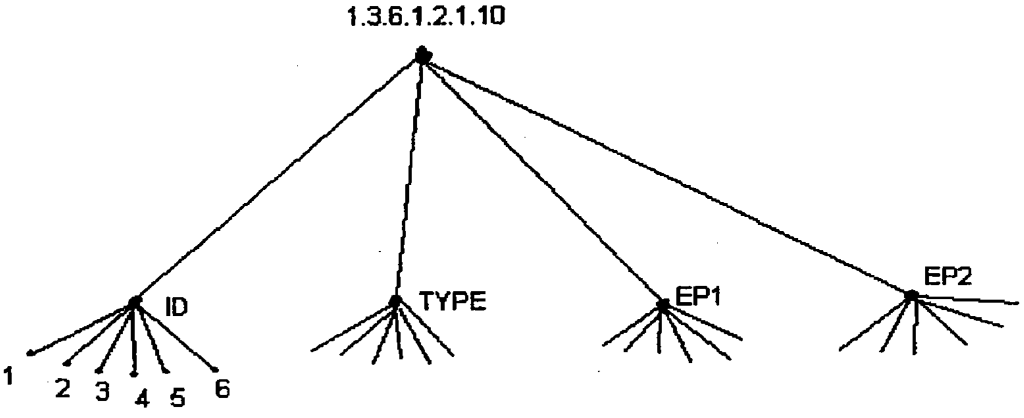


FIGURE 2A

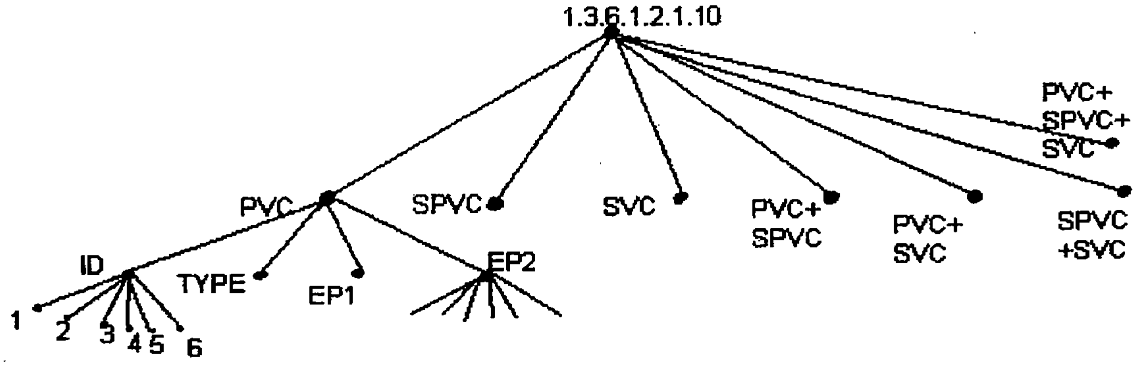


FIGURE 2B

Marks & Clerk

ID	Type	Endpoint 1	Endpoint 2
		Port/VPI/VCI	Port/VPI/VCI
1	SVCC	A/1/1	B/4/2
2	PVCC	A/2/1	B/1/1
3	SPVCC	A/3/1	C/4/0
4	SVCC	B/4/3	C/18/0
5	PVCC	B/4/4	C/20/4
6	SPVCC	B/4/5	C/15/0

FIGURE 3

ID	Type	Endpoint 1	Endpoint 2
		Port/VPI/VCI	Port/VPI/VCI
1	SVCC	A/1/1	B/4/2
3	SPVCC	A/3/1	C/4/0
4	SVCC	B/4/3	C/18/0
6	SPVCC	B/4/5	C/15/0

FIGURE 4

Marks & Clerk

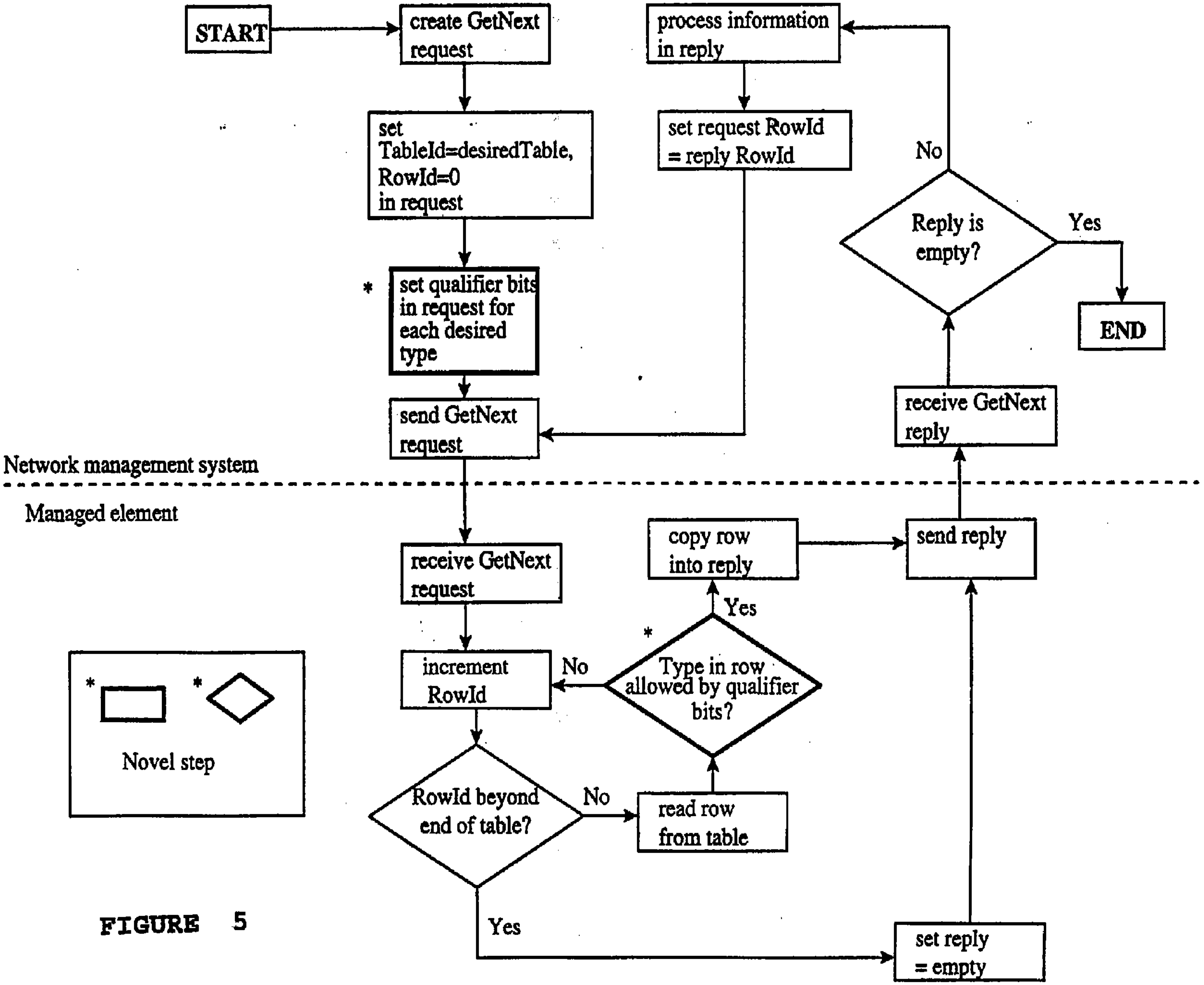

START
create GetNext request
set TableId=desiredTable, RowId=0 in request
*
set qualifier bits in request for each desired type
send GetNext request
process information in reply
set request RowId = reply RowId
No
Reply is empty?
Yes
END
receive GetNext reply
Network management system
Managed element
receive GetNext request
increment RowId
No
*
Type in row allowed by qualifier bits?
Yes
copy row into reply
send reply
RowId beyond end of table?
No
read row from table
Yes
set reply = empty
*
*
Novel step

FIGURE 5

Marks & Clerk

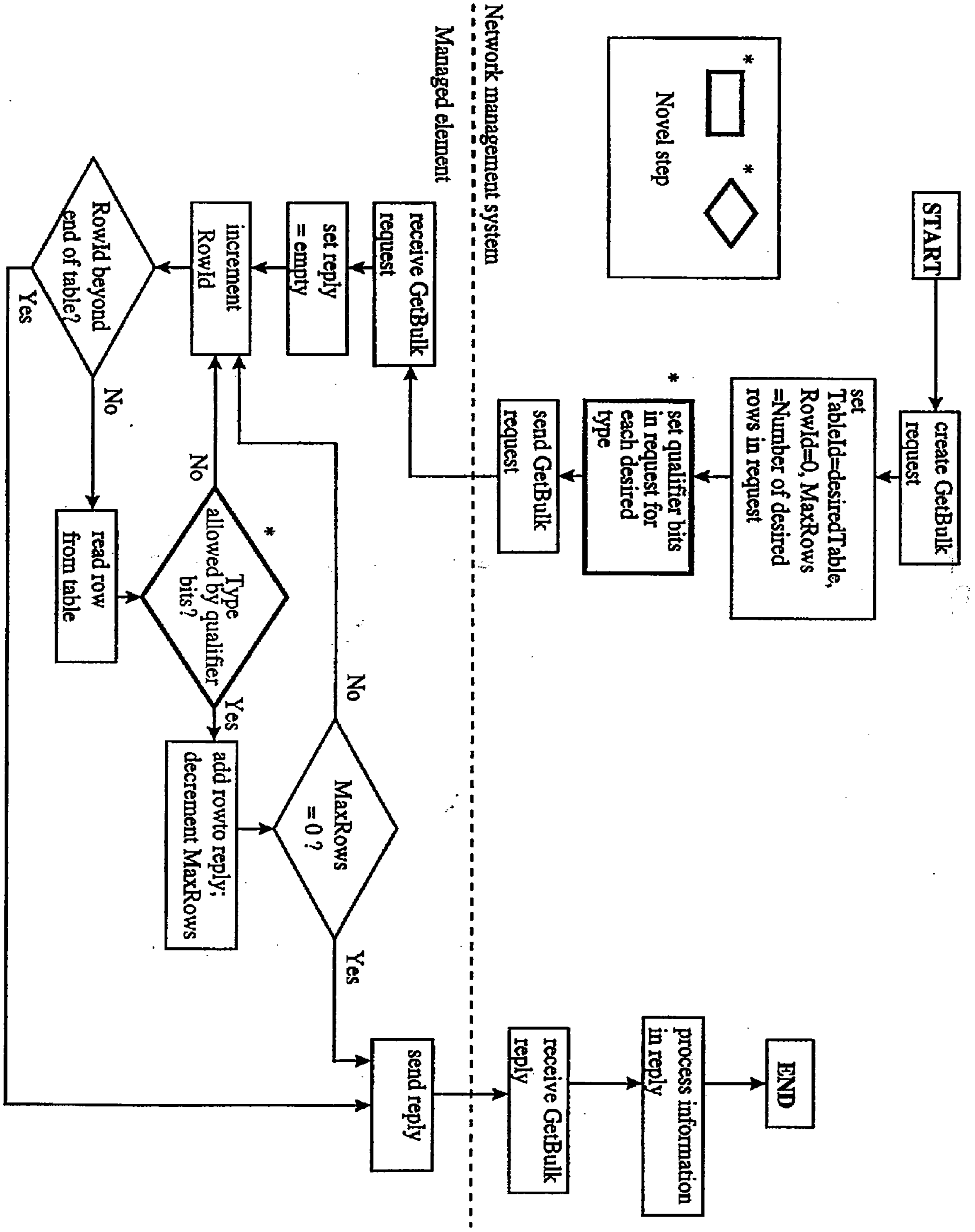

START
create GetBulk request
set TableId=desiredTable, RowId=0, MaxRows =Number of desired rows in request
*
set qualifier bits in request for each desired type
send GetBulk request
Novel step
*
*
Network management system
Managed element
receive GetBulk request
set reply = empty
increment RowId
RowId beyond end of table?
Yes
No
read row from table
*
Type allowed by qualifier bits?
Yes
No
add rowto reply; decrement MaxRows
MaxRows = 0 ?
Yes
No
send reply
receive GetBulk reply
process information in reply
END

FIGURE 6